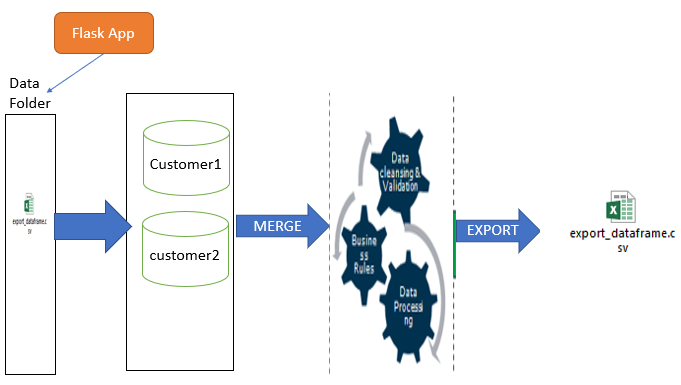
**SESSION M ETL solutions**

*Overview Of Solution –*



**Pre – Processing of Data –**

**Extraction** –

Data is Extracted from the both the CSV Files customer1.csv and customer2.csv from the Specified Location. Both CSV were passed to a pandas Dataframe and were assumed to be merged on the Basis of ‘ID’ Column as ‘Referral’ Codes as one customer code can be used by multiple Customers.

***Libraries Used –***

1)Numpy

2)Pandas

3)Phone numbers

4)Flask

5)email\_validator

***Python Files –***

*Session\_ETL.py –*

*This python file is used to upload and write the CSV files to the Destination folder Specified.*

*Pre\_process.py –*

*Program to Process CSV Files and produce the Output Files.*

*App.py*

*Configuration of Flask Application.*

***Functions –***

*valid\_email – To Validate Email if Correct ,email\_valid\_check column is created to show if email is valid or not*

*read\_csv – To Read Csv file from the path specified ,The Function Accepts runtime Argument path*

*etl\_preprocess – function Built to pre-process and Transform all 17 columns specified, Accepts customer1.Csv and Customer2.csv as argument from the path specified*

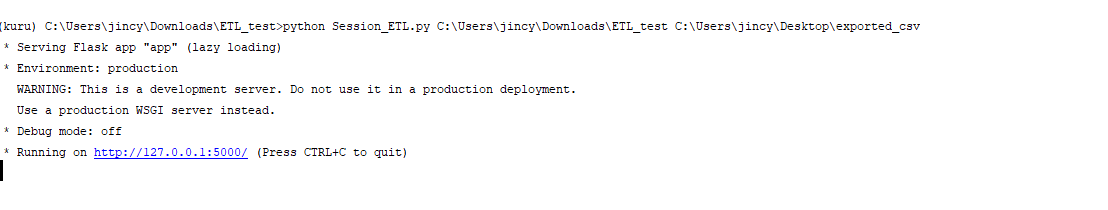
*Method of Execution –*

The Folder contains executable Session\_ETL.py files which can be executed to produce the Transformed Output files “export\_dataframe.csv” File.

The Entire Script is Deployed on Flask Application which can be called using the below Command

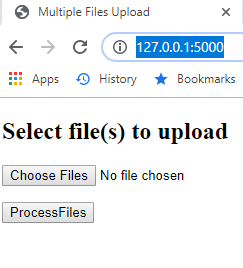
**CMD:**

python Session\_ETL.py *csv\_file\_path*  *Output\_file\_path*



Snap : *Once Executed Files Can be uploaded using the below address –*

<http://127.0.0.1:5000/>



On Successful completion of processing of csv output Files :



*MERGE condition –*

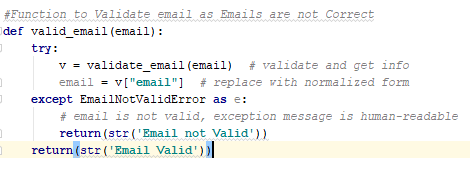
Assumptions -

|  |  |  |
| --- | --- | --- |
| **Merge Conditon** | **Similar Columns** | **Assumptions** |
| df\_temp = pd.merge(File2.loc[:, ['id', 'sex', 'tier', 'attr1', 'attr2','gender']], File1.iloc[:, 0:7], on=['id'],  how='left') | ID | ID Column was Assumed as 'Referral Code' as one referral Code can be associcated with multiple Customers |

Below Transformation steps gives all the list of Columns and their Derivation .

*Data Cleansing and Sanity checks-*

1)Email Column – Email column is Validated using the Below function to Give Custom Column EMAIL\_VALID\_CHECK Column is Either Email is Valid or Not



2) conversion of ‘Na’ values to Blank from the Dataset to prevent from typecasting.



3) Conversion of Attr1 column in customer2.csv files to lowercase as when validated both Contained Same value with different cases.



4)Mapping of Sex Column to produce Either ‘M’ or ‘F’





**Transformation/Assumptions/Questions** –

Pre-process Function was used to Derived all the Below columns –

|  |  |  |
| --- | --- | --- |
| **Columns Extracted** | **Dataype** | **Transformation** |
| External\_id | String | Derived from Email Columns if present for Customer otherwise NA |
| opted\_in | String | Checked If Loyalty customer by 'Tier' field of Customer |
| External\_id\_type | String | The Id column represent type of customer |
| Email | String | Email Column was used directly for the same |
| Locale | String | attr2 i.e. Phonenumber was used here to get the region of customer |
| ip | String | ? |
| dob | String | ? |
| Address | String | ? |
| City | String | The City of Customer was derived from the locale of Customer |
| State | string | ? |
| Zip | string | ? |
| Country | string | ? |
| Referral | String | Derived from Id Column as one Code of Friend could be used by multiple Customer |
| phone numbers | Jsonb | Derived from Phone number Colum and is of Default type 'Mobile' |
| Gender | string | Derived from Gender Mapping either 'M' or 'F' |
| First\_name | string | Derived from First\_name Column |
| Last\_name | string | Derived from Last\_name Column |
| Email Valid check | string | Gives info if Email is Valid or Not |

Output Files Produced –

